



TITLE:

# 疎性を持っている多項式最適化問題に対する半正定値計画緩和(最適化数理の手法と実際)

AUTHOR(S):

脇, 隼人; 小島, 政和; 村松, 正和

---

CITATION:

脇, 隼人 ...[et al]. 疎性を持っている多項式最適化問題に対する半正定値計画緩和(最適化数理の手法と実際). 数理解析研究所講究録 2005, 1461: 1-14

ISSUE DATE:

2005-12

URL:

<http://hdl.handle.net/2433/47954>

RIGHT:

# 疎性を持っている多項式最適化問題に対する 半正定値計画緩和

東京工業大学 大学院情報理工学研究科 脇 隼人 (Hayato Waki)

Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology

Department of Mathematics, Ewha Women's University Kim Sunyoung

東京工業大学 大学院情報理工学研究科 小島 政和 (Masakazu Kojima)

Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology

電気通信大学 情報工学科 村松 正和 (Masakazu Muramatsu)

Dept. of Computer Science, The University of Electro-Communications

## 1 はじめに

半正定値計画問題とは、線型計画問題を対称行列空間に拡張した最適化問題である。半正定値計画問題は、主双対内点法と呼ばれるアルゴリズムを用いて多項式時間で解くことができ、ソフトウェアも充実している [13, 15]。

この半正定値計画問題の応用例として、半正定値計画緩和がある。それは、与えられた最適化問題を半正定値計画問題に変形して、最適化問題の下界値を求める手法である。この半正定値計画緩和に関して様々な研究がなされているが、近年、Lasserre に [8] や Parrilo [11] よって、多項式最適化問題に対して半正定値計画緩和を利用する手法が提案された。この手法は、理論的には、多項式最適化問題の大域的最適値や最適解を得ることができる。

多項式最適化問題とは、 $n$  変数の実多項式  $f_0, f_j$  ( $j = 1, \dots, m$ ) で記述される以下の最適化問題のことである：

$$\text{最小化 } f_0(x) \quad \text{制約 } f_j(x) \geq 0 \quad (j = 1, \dots, m), x \in \mathbb{R}^n \quad (1)$$

多項式最適化問題は、組合せ最適化問題や 2 次計画問題など様々な最適化問題を記述することができる。

Lasserre による半正定値計画緩和は、理論的には強力な手法であるが、実際、変数の数が 15 を越えると、得られる半正定値計画問題が大規模なものになり解くことが困難になる。

本稿では、まず、Lasserre による半正定値計画緩和を紹介し、疎性をもった多項式最適化問題に対して、疎性を利用することで Lasserre の手法では解けないサイズの最適化問題を高速に解けることを述べる。なお、本稿は [14] に基づいている。

## 2 準備

この節では、多項式や多項式の 2 乗和 (Sums of Squares) について述べる。

まず、 $\mathbb{R}[x]$  を  $x = (x_1, \dots, x_n)^T$  で定義される多項式の集合とする。変数  $x$  と非負整数ベクトル  $a = (a_1, \dots, a_n)^T$  に対して、単項式  $x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$  を  $x^a$  と書く。これにより、多項式  $f$  は、単項式  $x^a$  とその単項式に対する係数  $c_a$  を用いて、

$$f(x) = \sum_{a \in \mathcal{F}} c_a x^a$$

と書くことができる。ここで、 $\mathcal{F} = \{a \in \mathbb{Z}_+^n \mid c_a \neq 0\}$  であり、多項式  $f$  のサポートと呼ぶ。さらに、 $f$  の次数は  $\deg(f) = \max\{\sum_{i=1}^n a_i \mid a \in \mathcal{F}\}$  である。

次に、多項式の 2 乗和に関して述べる。まず、 $\mathcal{P}$  を全ての  $x \in \mathbb{R}^n$  で非負の値をとる多項式の集合とし、 $\Sigma$  を次の様に定める：

$$\Sigma = \left\{ p \in \mathbb{R}[x] \mid p(x) = \sum_{k=1}^K q_k(x)^2, q_k \in \mathbb{R}[x] \right\}.$$

これらの集合には、 $\Sigma \subseteq \mathcal{P}$  という関係が成立することが知られている [12]。また、任意の正の整数  $r$  に対して、 $\Sigma_r = \{p \in \Sigma \mid \deg(p) \leq 2r\}$  と定めると、 $\Sigma_r \subset \Sigma$  であることがわかる。この  $\Sigma_r$  の要素に関して次の命題が成立することが知られている [11]：

**補題 2.1** 正の整数  $r$  と変数  $x$  に対して、

$$u_r(x) := (1, x_1, \dots, x_n, x_1^2, x_1x_2, \dots, x_n^2, \dots, x_n^r)^T$$

と定める。 $u_r(x)$  は次数  $r$  以下の全ての単項式で構成されている。この時、 $p \in \Sigma_r$  であることと、次の恒等式を満たす半正定値対称行列  $V$  が存在することは同値である：

$$p(x) = u_r(x)^T V u_r(x) \quad (\forall x \in \mathbb{R}^n).$$

### 3 制約式のない多項式最適化問題に対する半正定値計画緩和

この節では、制約式がない多項式最適化問題に対する半正定値計画緩和と、[7] で提案されている工夫について述べる。

制約式のない多項式最適化問題とは以下の最適化問題である：

$$\text{最小化 } f(x) \quad \text{制約 } x \in \mathbb{R}^n. \quad (2)$$

$f$  が奇数の次数の場合、最小値は  $-\infty$  になるので、 $\deg(f) = 2r$  の場合を考える。また次数が偶数でも  $f(x_1, x_2) = x_1x_2$  の様に最小値が  $-\infty$  になる場合があるので、この最小化問題 (2) の最小値  $f^*$  が有限値であると仮定する。この時、 $f^*$  が最小値であるということから、以下の最大化問題の最大値と  $f^*$  が一致する：

$$\text{最大化 } p \quad \text{制約 } f(x) - p \geq 0 \quad (\forall x \in \mathbb{R}^n). \quad (3)$$

次に、(3) に着目する。この最適化問題の決定変数は  $p$  であり、制約式は、「どの  $x$  を選んでも不等式  $f(x) - p \geq 0$  を満たす  $p$  が実行可能解」、ということを意味している。これは、 $f(x) - p$  を  $x$  の多項式で  $p$  を定数とみなしたときに、「 $f(x) - p$  が  $\mathbb{R}^n$  で非負の値をとる多項式になる  $p$  が実行可能解」と見ることができる。つまり、(3) は以下の最適化問題と等価になる：

$$\text{最大化 } p \quad \text{制約 } f(x) - p \in \mathcal{P}. \quad (4)$$

この最適化問題の  $\mathcal{P}$  を  $\Sigma_r$  で置き換えて以下の最大化問題を得る：

$$\text{最大化 } p \quad \text{制約 } f(x) - p \in \Sigma_r. \quad (5)$$

なお,  $\Sigma_r \subseteq \mathcal{P}$  であるから, 最大値に関して

$$((2) \text{ の最小値}) = ((4) \text{ の最大値}) \geq ((5) \text{ の最大値}) \quad (6)$$

となっている。ここで, 最適化問題 (5) に着目する。補題 2.1 を利用すると, 以下のように書き換えることができる:

$$\text{最大化 } p \quad \text{制約 } f(x) - p = u_r(x)^T V u_r(x) \quad (\forall x \in \mathbb{R}^n), \quad V \succeq O. \quad (7)$$

ここで,  $V \succeq O$  は  $V$  が半正定値行列であることを意味する。最適化問題 (7) の制約式にある等式は  $x$  に関する恒等式なので, 両辺を比較することで,  $V$  に関する線形方程式を得ることができる。そこで比較するために, 恒等式の右辺を変形すると,

$$u_r(x)^T V u_r(x) = \text{Tr}(V u_r(x) u_r(x)) = E_0 + \sum_{a \in \mathcal{A}_{2r} \setminus \{0\}} \text{Tr}(V E_a)$$

と書くことができる。ここで,  $E_0$  は  $(1, 1)$  成分が 1 で残りが 0 の正方行列で,  $E_a$  も同様に定数行列であり,  $\mathcal{A}_r = \{a \in \mathbb{Z}_+^n \mid \sum_{j=1}^n a_j \leq r\}$  である。

例 3.1  $n=2, r=2$  の場合,  $u_2(x_1 x_2) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)^T$  であり,

$$u_r(x) u_r(x)^T = \begin{pmatrix} 1 & x_1 & x_2 & x_1^2 & x_1 x_2 & x_2^2 \\ x_1 & x_1^2 & x_1 x_2 & x_1^3 & x_1^2 x_2 & x_1 x_2^2 \\ x_2 & x_1 x_2 & x_2^2 & x_1^2 x_2 & x_1 x_2^2 & x_2^3 \\ x_1^2 & x_1^3 & x_1^2 x_2 & x_1^4 & x_1^3 x_2 & x_1^2 x_2^2 \\ x_1 x_2 & x_1^2 x_2 & x_1 x_2^2 & x_1^3 x_2 & x_1^2 x_2^2 & x_1 x_2^3 \\ x_2^2 & x_1 x_2^2 & x_2^3 & x_1^2 x_2^2 & x_1 x_2^3 & x_2^4 \end{pmatrix}$$

となる。これより, 例えば,

$$E_{(1,1)} = \begin{pmatrix} & & 1 \\ & 1 & \\ 1 & & \end{pmatrix}, E_{(2,1)} = \begin{pmatrix} & & 1 \\ & 1 & \\ 1 & & \end{pmatrix}, E_{(2,2)} = \begin{pmatrix} & 1 & \\ & & 1 \\ & & 1 \end{pmatrix},$$

となる。この例では, 全ての要素が 0, 1 になっているが,  $n$  や  $r$  がどんな値でも,  $E_a$  の要素は 0, 1 である。

したがって, 恒等式の両辺を比較することで, 以下の最適化問題を得る:

$$\text{最大化 } p \quad \text{制約 } \begin{cases} -p = \text{Tr}(V E_0), & c_a = \text{Tr}(V E_a), \quad a \in \mathcal{A}_{2r} \setminus \{0\}, \\ V \succeq O. \end{cases} \quad (8)$$

この最適化問題 (8) は, 半正定値計画問題になっており, 主双対内点法を適用することで, この問題 (8) の最大値を得ることができる。補題 2.1 を利用した最後の変形は等価な変形なので, (6) の関係から,

$$((2) \text{ の最小値}) \geq ((8) \text{ の最大値})$$

となる。このことから、(8) を解くことで、(2) の最小値の下限を求めることができる。

しかしながら、半正定値計画問題 (8) における変数  $V$  は、 $\binom{n+r}{r} \times \binom{n+r}{r}$  の対称行列であり、等式制約は、 $\binom{n+2r}{n}$  あるので、 $n$  や  $r$  が大きくなると、(8) のサイズが大規模になり、実用上、最適値を求めることが困難になる。

この困難を克服するために、[7] では、 $f$  の単項式の数、すなわち、 $f$  のサポート  $\mathcal{F}$  の要素数が  $\mathcal{A}_{2r}$  の要素と比べて少ない多項式に対して、得られる半正定値計画問題の規模を小さくする手法が提案されている。この手法は、 $\mathcal{F}$  の情報から、 $f(x) - p$  を多項式に 2 乗和にするのに必要のない単項式を  $\mathcal{A}_r$  から除くものである。そのようにして得られる単項式の集合を  $\mathcal{G}^*$  とすると、

$$f(x) - p = u_r(x; \mathcal{G}^*)^T V u_r(x; \mathcal{G}^*)$$

の両辺を比較することで、得られる半正定値計画問題の規模が小さくなるのである。ここで、 $u_r(x; \mathcal{G}^*)$  は縦ベクトルで、 $\mathcal{G}^* = \{a_1, \dots, a_K\}$  とすると、 $u_r(x; \mathcal{G}^*) = (x^{a_1}, \dots, x^{a_K})^T$  である。この議論により、 $\mathcal{G}^*$  という集合より大きく単項式の集合をとる必要がないことが分かる。

しかしながら、[7] でも言及されているように、この手法では必ずしも半正定値計画問題のサイズを小さくできるわけではない。例えば、 $f(x) = \sum_{i=1}^n x_i^{2r}$  の様な多項式を最小化する場合、 $\mathcal{G}^* = \mathcal{A}_r$  となり、得られる半正定値計画問題の規模は変わらない。

#### 4 疎性を利用した半正定値計画緩和

前節では、(2) から得られる半正定値計画問題が大規模になることを述べた。この節では、扱っている多項式  $f$  の構造を利用して、得られる半正定値計画問題を小さくすることについて述べる。

$n$  変数多項式  $f$  に対して、以下の様に行列  $S \in \mathbb{R}^{n \times n}$  を構成する：

$$S_{ij} = \begin{cases} * & (i = j), \\ * & a \in \mathcal{F} \text{ に対して } a_i \geq 1 \text{ かつ } a_j \geq 1, \\ 0 & \text{それ以外.} \end{cases}$$

$*$  は非ゼロ要素であることを表している。この行列  $S$  が疎行列のとき、 $f$  を疎な多項式と呼ぶことにする。

例 4.1  $f(x) = \sum_{i=1}^{n-1} (x_i^4 + 2x_i x_{i+1}^2 - x_{i+1}^2)$  の場合、 $S$  は、

$$S = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & & n \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ \\ n \end{matrix} & \begin{pmatrix} * & * & & & \\ * & * & * & & \\ & * & * & \ddots & \\ & & \ddots & \ddots & * \\ & & & * & * \end{pmatrix} \end{matrix}$$

となる。

次に、この  $S$  に対して、グラフ理論の結果を利用する。そのために、 $S$  から無向グラフ  $G = (V, E)$  を以下の様に構成する：

$$\text{頂点集合 } V = \{1, 2, \dots, n\},$$

$$\text{辺集合 } E = \{(i, j) \mid i, j \in V, S_{ij} = \star\}.$$

この様にできた無向グラフ  $G$  に対して、極大クリーク集合族  $\mathcal{C} = \{C_1, \dots, C_\ell\}$  を取り出す。これらのクリーク  $C_k \subset V$  ( $k = 1, \dots, \ell$ ) に対して  $\mathcal{A}_r^{C_k} = \{a \in \mathcal{A}_r \mid a_i = 0, \forall i \notin C_k\}$  と定めて、

$$f(x) - p = \sum_{k=1}^{\ell} u_r(x; \mathcal{A}_r^{C_k})^T V_k u_r(x; \mathcal{A}_r^{C_k})$$

という恒等式を構成する。今、 $C_k$  をクリークとしているので、必ず任意の  $a \in \mathcal{F}$  に対して  $x^a$  が  $\mathcal{A}_{2r}^{C_k}$  に属している。したがって、 $f$  の単項式  $x^a$  は必ず、恒等式の右辺にも現れる。この恒等式に対して、両辺を比較することで以下の半正定値計画問題を得ることができる：

$$\text{最大化 } p \quad \text{制約} \quad \begin{cases} c_a = \sum_{k=1}^{\ell} \text{Tr}(V_k E_{k,a}), & a \in \bigcup_{k=1}^{\ell} \mathcal{A}_{2r}^{C_k} \setminus \{0\}, \\ -p = \sum_{k=1}^{\ell} \text{Tr}(V_k E_{k,0}), & V_1, \dots, V_\ell \succeq 0 \end{cases} \quad (9)$$

なお、 $E_{k,a}$  は、行列  $u_r(x; \mathcal{A}_r^{C_k}) u_r(x; \mathcal{A}_r^{C_k})^T$  の  $x^a$  に対応する係数行列であり、0 と 1 で構成されている。

例 4.2  $f(x) = \sum_{i=1}^{n-1} (x_i^4 + 2x_i x_{i+1}^2 - x_{i+1}^2)$  に対して、例 4.1 で求めた  $S$  より、クリーク  $C_k$  は

$$C_k = \{k, k+1\}, \quad (k = 1, \dots, n-1)$$

となる。したがって、 $\deg(f) = 4$  なので  $r = 2$  となり、

$$u_r(x; \mathcal{A}_r^{C_k}) = u_2(x; \mathcal{A}_2^{C_k}) = (1, x_k, x_{k+1}, x_k^2, x_k x_{k+1}, x_{k+1}^2)^T$$

となる。このことから、得られる半正定値計画問題 (9) の変数行列  $V_k$  のサイズは  $6 \times 6$  であり、

$$\bigcup_{k=1}^{\ell} \mathcal{A}_{2r}^{C_k} = \{0, x_1, \dots, x_n, x_1^2, x_2^2, \dots, x_n^2, x_1 x_2, x_2 x_3, \dots, x_k x_{k+1}, \dots, x_{n-1} x_n\}$$

なので、等式の数、 $3n$  となり、工夫せずに得られる半正定値計画問題 (8) の規模と比べると、問題の規模が減少していることが分かる。

しかしながら、要素数が最大のクリークを見つけるのでさえ NP 困難なので、極大クリークを全て見つけて、恒等式を構成することが難しいことが分かる。そこで、無向グラフ  $G$  から、辺を付け足すことでコーダルグラフを作り、そのコーダルグラフから極大クリーク族を

見つけるようにする。これは、コーダルグラフから容易に極大クリーク族を見つけていることができるからである [3]。

なお、無向グラフ  $G$  に辺を付け足すとあるが、これに関しても、付け足す辺の数を最小にすることは NP 完全であることが知られているので、コレスキー分解を利用している。詳細は、[4, 14] を見て欲しい。

上の例 4.2 で示しているように、極大クリーク族を利用することで得られる半正定値計画問題 (9) の規模が小さくなっていることが分かるが、一方、この様に規模を小さくすることは、

$$\Sigma'_{2r} = \left\{ g \in \Sigma_{2r} \mid g(x) = \sum_{k=1}^{\ell} g_k(x)^2, \quad g_k \text{ のサポートは } A_r^{C_k} \right\}$$

とおけば、(5) において、 $\Sigma_{2r}$  の代わりに  $\Sigma'_{2r}$  を用いていることになる。したがって、 $\Sigma'_{2r} \subset \Sigma$  が成立するので、

$$((9) \text{ の最大値}) \leq ((8) \text{ の最大値}) \leq ((2) \text{ の最小値})$$

となる。つまり、元の最小化問題 (2) の下界値を求める半正定値計画問題 (8) よりも悪い下界値を得る可能性があるということである。

## 5 制約式のある多項式最適化問題に対する半正定値計画緩和

この節では、まず、(1) に対してどのように半正定値計画緩和 [8] を適用するかを [6] に基づいて議論する。次に、前節で提案した手法をどの様に用いるかについて述べ、最後に、得られる半正定値計画問題の双対問題に着目する。

### 5.1 半正定値計画緩和について

制約式がある場合は、制約式がない最適化問題に変形して前節で説明した手法で半正定値計画問題を導く。そこで、(1) に対して、一般化 Lagrange 関数  $L(x, \psi)$  を定義する。任意の  $x \in \mathbb{R}^n, \psi \in \Sigma^m$  に対して、

$$\tilde{L}(x, \psi) = f_0(x) - \sum_{j=1}^m f_j(x) \psi_j(x)$$

ここで  $\Sigma^m$  は前節で定義した多項式の集合  $\Sigma$  の  $m$  個の直積である。 $\mathbb{R}^m$  の非負領域  $\mathbb{R}_+^m$  は、 $\Sigma^m$  の部分集合なので、 $\psi$  として  $\mathbb{R}_+^m$  の要素を選べば、 $L(x, \psi)$  は、最適化の分野で良く用いられる Lagrange 関数に対応している。

$\psi$  を固定して、この一般化 Lagrange 関数を最小化する以下の最適化問題を考える：

$$\text{最小化 } L(x, \psi) \quad \text{制約 } x \in \mathbb{R}^n.$$

この最小化問題の最小値を  $p^*(\psi)$  とすると、任意の  $\psi \in \Sigma^m$  に対して、

$$p^*(\psi) \leq ((1) \text{ の最小値}),$$

が成立する。このことから、以下の不等式が成立することが分かる：

$$\max_{\psi \in \Sigma^m} \min_{x \in \mathbb{R}^n} L(x, \psi) = \max_{\psi \in \Sigma^m} p^*(\psi) \leq ((1) \text{ の最小値}). \quad (10)$$

実は、[6] では、 $\psi$  として、(1) の実行可能領域に対する罰金関数を選ぶことで、(10) における不等号は等式に置き換わることを示している。

以上の議論より、(10) の左辺の最適化問題に、半正定値計画緩和を施すことで (1) の下界値または最小値を得ることができる。

次に、(10) の左辺の最適化問題に対する半正定値計画緩和について述べる。まず、 $\psi$  を固定して以下の最適化問題を考える：

$$\text{最小化 } L(x, \psi) \quad \text{制約 } x \in \mathbb{R}^n.$$

この最適化問題は、次の最適化問題と同じ最適値をとる：

$$\text{最大化 } p \quad \text{制約 } L(x, \psi) - p \geq 0, \quad (\forall x \in \mathbb{R}^n).$$

この最適化問題において変数は  $p$  のみで、制約は、「どの  $x$  を選んでも不等式を満たす  $p$  が実行可能解」、ということの意味している。

この最適化問題の制約に着目すると、これは、 $\psi$  が固定されているので、 $x$  に関する多項式  $L(x, \psi) - p$  が全ての  $x$  で非負の値をとる、とみなすことができる。つまり、 $L(x, \psi) - p \in P$  である。今、ここで、 $P$  をより狭い集合  $\Sigma$  に置き換え、次の最適化問題を得る：

$$\text{最大化 } p \quad \text{制約 } L(x, \psi) - p \in \Sigma.$$

ここで、 $\psi$  を動かせば、

$$\text{最大化 } p \quad \text{制約 } L(x, \psi) - p \in \Sigma, \quad \psi = (\psi_1, \dots, \psi_m) \in \Sigma^m.$$

となる。ここで、補題 2.1 を利用するためには、 $\Sigma$  や  $\Sigma^m$  の要素に対して次数を制限しなくてはならない。そのため、 $L(x, \psi) - p$  の次数が  $2r$  であると制限する。これにより、 $2r$  になるように  $\psi_i$  の次数を制限できる。なお、この  $r$  は、 $r \geq \lceil \max\{\deg(f_j)\}/2 \rceil$  を満たす整数であれば自由にとることができる。 $L(x, \psi) - p$  の次数が  $2r$  になるように、 $\psi_j$  の次数を  $r_j$  と制限して、補題 2.1 を適用すれば、この最適化問題は、以下のようになる：

$$\text{最大化 } p \quad \text{制約 } \begin{cases} L(x, \psi) - p = u_r(x)^T W u_r(x), & (\forall x \in \mathbb{R}^n), W \succeq O \\ \psi_j(x) = u_{r_j}(x)^T V_j u_{r_j}(x), & (j = 1, \dots, m, \forall x \in \mathbb{R}^n), V_j \succeq O \end{cases}$$

この最適化問題において、 $\psi_j$  を  $L(x, \psi)$  に代入すれば、

$$\text{最大化 } p \quad \text{制約 } \begin{cases} f(x) - \sum_{j=1}^m f_j(x) u_{r_j}(x)^T V_j u_{r_j}(x) - p = u_r(x)^T W u_r(x), & (\forall x \in \mathbb{R}^n) \\ W, V_j \succeq O. \end{cases} \quad (11)$$

を得る。なお、この最適化問題では、 $W, V_j, p$  が変数である。この最適化問題の制約は  $x$  に関する恒等式なので、両辺を比較すると、 $W, V_j, p$  に関する線形方程式を得る。したがって、(1) の半正定値計画緩和を施すことによって、以下の半正定値計画問題を得る：

$$\text{最大化 } p \quad \text{制約 } \begin{cases} W, V_j, p \text{ に関する線形方程式系,} \\ W, V_j \succeq O \end{cases} \quad (12)$$



この最適化問題は,  $r$  を 1 つ定めることで, 半正定値計画問題 (12) を 1 つ構成することができる. この半正定値計画問題を得るために,  $\mathcal{P}$  を  $\Sigma$  に,  $\Sigma$  を  $\Sigma_r$  や  $\Sigma_{r_j}$  という部分集合に置き換えたので, (12) の最大値を  $p_r$ , (1) の最小値を  $p^*$  とおくと,

$$p_r \leq p^* \quad (\forall r)$$

という不等式が成立する. また,  $\Sigma_r \subset \Sigma_{r+1}$  という関係が成立するので,

$$p_r \leq p_{r+1} \quad (\forall r)$$

が成立することがわかる. 実は, [8] では, ある仮定を満たすとき, 次の式が成り立つことが示されている:

$$\lim_{r \rightarrow \infty} p_r = p^*$$

また, [8] ではいくつかのテスト問題 [2] に対して  $r = 2, 3$  で最適値が得られていることを紹介している.

このように, 半正定値計画緩和は, 理論的には非常に強力であるが, 一方で, 変数の数が 15 を越えると得られる半正定値計画問題が大規模になり解くのに非常に時間がかかるので, 大規模な多項式最適化問題 (1) に対しては, 半正定値計画緩和がうまく働かないことが想像できる. そこで, 前節で提案した, 疎性を利用した半正定値計画緩和を適用して, 高速に最小値やその下界値が求められることを次節で述べる.

## 5.2 疎性を利用した半正定値計画緩和

この節では, 疎性を持った多項式最適化問題 (1) に対して, 疎性を利用することで, 半正定値計画緩和で得られる半正定値計画問題よりサイズの小さい半正定値計画問題を導けることを述べる. なお, この節の詳細は [6] にある.

半正定値計画問題のサイズが大きくなった理由は,  $u_r(x)$  の要素数が多くなり,  $W, V_j$  のサイズが大きくなるからである. そこでまず,  $\Sigma_{r_j}$  の部分集合  $\widetilde{\Sigma}_{r_j}$  を以下の様に定義し,  $\psi_j \in \Sigma_{r_j}$  を  $\psi_j \in \widetilde{\Sigma}_{r_j}$  に置き換えて半正定値計画問題を導く:

$$\widetilde{\Sigma}_{r_j} = \left\{ p \in \Sigma_r \mid p = \sum_{k=1}^K q_k(x)^2, q_k \text{ のサポートは } \mathcal{F}_j^{F_j} \right\}$$

ここで,  $F_j$  は,  $f_j$  のサポート  $\mathcal{F}_j$  に対して,  $F_j = \{k \mid \text{ある } a \in \mathcal{F}_j \text{ に対して } a_k \geq 1\}$  と定めた添字集合である. 例えば,  $f_1(x) = 1 - x_1^2 - x_2x_5$  ならば,

$$\mathcal{F}_1 = \{(0, 0, 0, 0, 0), (2, 0, 0, 0, 0), (0, 1, 0, 0, 1)\}$$

であるから,  $F_1 = \{1, 2, 5\}$  である. このように制限することで,  $V_j$  のサイズを小さくすることができる. しかし,  $L(x, \psi) - p$  に対するこの  $F_j$  ような添字集合は,  $\{1, 2, \dots, n\}$  となるので, 同様に  $W$  のサイズを小さくすることはできない. そこで, 疎性を利用してサイズを小さくする方法を述べる. まず, (1) に対する行列  $S$  を以下の様に定義する:

$$S_{ij} = \begin{cases} * & i = j; \\ * & a \in \mathcal{F}_0 \text{ に対して } a_i, a_j \geq 1, \\ * & \text{ある } \ell \text{ に対して } i, j \in F_\ell, \\ 0 & \text{それ以外.} \end{cases}$$

この様に定義された  $S$  が疎行列のとき, (1) は疎であると呼ぶことにする.

この  $S$  に対して無向グラフ  $G$  を構成し, 極大クリーク族を  $\mathcal{C} = \{C_1, \dots, C_\ell\}$  とする. これを用いて, (11) に現れる恒等式を, 以下の恒等式に置き換える:

$$f_0(x) - \sum_{j=1}^m f_j(x) u_{r_j}(x: A_{r_j}^{F_j})^T V_j u_{r_j}(x: A_{r_j}^{F_j}) - p = \sum_{k=1}^{\ell} u_r(x: A_r^{C_k})^T W_k u_r(x: A_r^{C_k}), \quad (\forall x \in \mathbb{R}^n)$$

この両辺を比較することで, 行列  $W_k, V_j$  と  $p$  に関する線形方程式系を得ることができる. 以上より, 得られる半正定値計画問題は,

$$\text{最大化 } p \quad \text{制約} \quad \begin{cases} W_k, V_j, p \text{ の線型方程式系,} \\ W_k, V_j \succeq O \end{cases} \quad (13)$$

となる.

この半正定値計画問題 (13) を導くために,  $\Sigma_r$  の部分集合  $\widetilde{\Sigma_{r_j}}$  や極大クリーク族  $\mathcal{C}$  を用いて半正定値計画問題を構成しているので, 半正定値計画問題 (13) の最大値を  $\tilde{p}_r$  とすると, 任意の  $r$  に対して,

$$\tilde{p}_r \leq \widetilde{p_{r+1}} \quad \text{や} \quad \tilde{p}_r \leq p_r \leq p^*$$

が成立するが,

$$\lim_{r \rightarrow \infty} \tilde{p}_r = p^*$$

が成立することは, わからない.

### 5.3 半正定値計画緩和で導かれる問題の双対問題

半正定値計画問題 (12) や (13) の双対問題が, 妥当不等式や線形化という技術を利用することで導出されることを以下で述べる.

まず, 多項式最適化問題 (1) に対して, 妥当不等式を追加する. 妥当不等式とは,  $\mathbb{R}^n$  全体または, 考えている実行可能領域全体で成立する冗長な不等式であり, 例えば  $u_r(x)u_r(x)^T \succeq O$  は全ての  $x \in \mathbb{R}^n$  で成立するので, 妥当不等式である. さらに,  $f_j(x) \geq 0$  を満たす  $x$  は,  $f_j(x)u_{r_j}(x)u_{r_j}(x)^T \succeq O$  も満たすので, 妥当不等式である. そこで, これらを多項式最適化問題 (1) に追加する. これにより得られる最適化問題は以下の様になる:

$$\text{最小化 } f_0(x) \quad \text{制約} \quad \begin{cases} f_j(x)u_{r_j}(x)u_{r_j}(x)^T \succeq O, & (j = 1, \dots, m) \\ u_r(x)u_r(x)^T \succeq O \end{cases}$$

この最適化問題の実行可能領域は (1) と同じなので, この最適化問題の最小値と, (1) の最小値は一致している. なお,  $r_j$  は,  $f_j(x)u_{r_j}(x)u_{r_j}(x)^T \succeq O$  に現れる単項式の最大次数が  $2r$  以下になるように,  $r_j$  を定める.

次に, 単項式  $x^a$  を  $y_a$  と置き換える. これを線形化と呼ぶ. 例えば,  $u_r(x)u_r(x)^T$  は,

$$u_r(x)u_r(x)^T = E_0 + \sum_{a \in \mathcal{A}_{2r} \setminus \{0\}} E_a x^a$$

なので、線形化すると、

$$E_0 + \sum_{a \in \mathcal{A}_{2r} \setminus \{0\}} E_a y_a$$

となる。同様に、 $f_0(x) = \sum_{a \in \mathcal{F}_0} c_a x^a$  に対しては、 $\sum_{a \in \mathcal{F}_0} c_a y_a$  である。(5.3) に対して線形化することによって、以下の最適化問題を得る：

$$\text{最小化 } \sum_{a \in \mathcal{F}_0} c_a y_a \quad \text{制約} \quad \begin{cases} D_{j,0} + \sum_{a \in \mathcal{A}_{2r} \setminus \{0\}} D_{j,a} y_a \geq 0, & (j = 1, \dots, m) \\ E_0 + \sum_{a \in \mathcal{A}_{2r} \setminus \{0\}} E_a y_a \geq 0, & x^a = y_a, \quad (\forall a \in \mathcal{A}_{2r}) \end{cases} \quad (14)$$

ここで、 $D_{j,0}$  や  $D_{j,a}$  は、 $f_j(x) u_{r_j}(x) u_{r_j}(x)^T$  を線形化して得られる定数行列である。

最後に、 $x^a = y_a$  という制約を除くことで、(12) の双対問題を導くことができる。

また、 $u_r(x) u_r(x)^T \succeq 0$  を  $u_r(x: \mathcal{A}_r^{C_k}) u_r(x: \mathcal{A}_r^{C_k})^T \succeq 0$ ,  $(k = 1, \dots, \ell)$  で置き換え、 $f_j(x) u_{r_j}(x) u_{r_j}(x)^T \succeq 0$  を  $f_j(x) u_{r_j}(x: \mathcal{A}_{r_j}^{F_j}) u_{r_j}(x: \mathcal{A}_{r_j}^{F_j})^T \succeq 0$  で置き換えることで、線形化することで得られる半正定値計画問題は、半正定値計画問題 (13) の双対問題になっている。

## 6 数値実験

この節では、テスト問題 [5] に対して Lasserre によって提案された半正定値計画緩和と、疎性を利用した半正定値計画緩和をそれぞれ適用することで、これらの計算効率を比較する。そのために、まず、得られた最適値の精度や (12) や (13) の最適解からどの様に (1) の最適解の候補を取り出す方法を述べる。次に、数値実験の結果を述べる。なお、詳細は [14] を見て欲しい。

### 6.1 得られた最適値の精度について

まず、どのように、(12) や (13) の最適解から (1) の最適解の候補を取り出すかについて述べる。(1) の目的関数  $f_0(x)$  に対して、十分小さい  $\varepsilon > 0$  に対して、 $\|p\|_1 \leq \varepsilon$  を満たす  $p \in \mathbb{R}^n$  を用いて、 $f_0(x) + p^T x$  と置き換えて、半正定値計画緩和を適用する。これにより、(1) の最適解が一意になり、得られた半正定値計画問題の最適解  $y^*$  に対して  $\tilde{x} = (y_{e_1}^*, \dots, y_{e_n}^*)^T$  とすると、 $\tilde{x}$  は、 $f_0(x) + p^T x$  を目的関数とする (1) の最適解の候補となり、 $f_0(\tilde{x}) + p^T \tilde{x}$  と半正定値計画問題の最適値が等しければ、 $\tilde{x}$  を  $f_0(x) + p^T x$  を目的関数とする (1) の最適解とみなすことができる [14]。

次に、このベクトル  $\tilde{x}$  を  $f_0, f_j$  に対して以下のように定める：

$$\begin{aligned} \epsilon_{\text{obj}} &= \frac{|f_0(\tilde{x}) + p^T \tilde{x} - \text{半正定値計画問題の最適値}|}{\max 1, |f_0(\tilde{x}) + p^T \tilde{x}|} \\ \epsilon_{\text{feas}} &= \min_{1 \leq j \leq m} \{f_j(\tilde{x})\} \end{aligned}$$

また、もし (1) が等式制約  $h_k(x) = 0$  ( $k = 1, \dots, s$ ) を持っていれば、

$$\epsilon_{\text{feas}} = \min \{f_j(\tilde{x}) \quad (j = 1, \dots, m), -|h_k(\tilde{x})| \quad (k = 1, \dots, s)\}$$

とする. この時,  $\epsilon_{\text{obj}}$  と  $\epsilon_{\text{feas}}$  が十分 0 に近ければ,  $\tilde{x}$  は目的関数を  $f_0(x) + p^T x$  として (1) の最適解と認識することができる. ここで,  $p$  は十分ノルムが小さくなるようにとっているの  
で,  $\tilde{x}$  は (1) の最適解に近い, 近似解と見ることができる.

## 6.2 数値実験の結果

この数値実験では,  $\|p\|_1 < 1.0 * 10^{-5}$  とした. また, よりよい近似解を得るために,  $F_i$  ではなく,  $F_j$  を含むようなクリーク  $C_{i_1}, \dots, C_{i_q}$  を用いて,  $\cup_{k=1}^q C_{i_k}$  を利用して,

$$\widetilde{\Sigma_{r_j}} = \left\{ p \in \Sigma_r \mid p = \sum_{\ell=1}^K q_\ell(x)^2, q_\ell \text{ のサポートは } A_{r_i}^{\cup_{k=1}^p C_{k_j}} \right\}$$

とした.

さらに, [7] で議論されている手法と, スケーリングを行って, 数値的不安定をできるだけ起こさないようにしている [14].

次に制約式がない場合 (2) と制約式がある場合 (1) に分けて数値実験の結果を述べることにする. なお利用した計算機の CPU は 2.4GHz AMD Opteron でメモリーが 8.0GB である. また, 半正定値計画問題を解くために, SeDuMi 1.05[13] を利用している.

## 6.3 制約式がない場合

テスト問題として, 以下の関数を利用した:

- Broyden tridiagonal 関数 [9]

$$f_0(x) = ((3 - 2x_1)x_1 - 2x_2 + 1)^2 + \sum_{i=2}^{n-1} ((3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1)^2 + ((3 - 2x_n)x_n - x_{n-1} + 1)^2.$$

- chained Wood 関数 [1]:

$$f_0(x) = 1 + \sum_{i \in J} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 + 90(x_{i+3} - x_{i+2}^2)^2 + (1 - x_{i+2})^2 + 10(x_{i+1} + x_{i+3} - 2)^2 + 0.1(x_{i+1} - x_{i+3})^2),$$

ただし,  $J = \{1, 3, 5, \dots, n-3\}$  で  $n$  は 4 の倍数.

- generalized Rosenbrock 関数 [10]:

$$f_0(x) = 1 + \sum_{i=2}^n \left\{ 100(x_i - x_{i-1}^2)^2 + (1 - x_i)^2 \right\}.$$

これらの関数に対して半正定値計画問題を適用した結果を以下の表 1 にまとめる. なお, 表中の cl.str は, 行列  $S$  から得られるクリークの構造を表し, 例えば,  $4 * 3 + 5 * 2$  は, 要素数 4 のクリークが 3 と, 要素数 5 のクリークが 2 つ生成されたことを意味している. 表 1 から分かる通り, 高速に (2) の最適値が得られていることがみてとれる.

また, 次の関数を利用して, 疎性を利用しない半正定値計画緩和と比較する:

| $n$  | Broyden tridiagonal |                         |      | Chained Wood |                         |     | Generalized Rosenbrock |                         |     |
|------|---------------------|-------------------------|------|--------------|-------------------------|-----|------------------------|-------------------------|-----|
|      | cl.str              | $\epsilon_{\text{obj}}$ | cpu  | cl.str       | $\epsilon_{\text{obj}}$ | cpu | cl.str                 | $\epsilon_{\text{obj}}$ | cpu |
| 600  | 3*598               | 9.1e-7                  | 9.3  | 2*599        | 1.4e-5                  | 0.9 | 2*599                  | 6.2e-9                  | 3.4 |
| 700  | 3*698               | 9.0e-7                  | 10.9 | 2*699        | 1.6e-5                  | 1.1 | 2*699                  | 7.5e-9                  | 4.0 |
| 800  | 3*798               | 2.2e-7                  | 12.6 | 2*799        | 1.8e-5                  | 1.3 | 2*799                  | 3.5e-9                  | 5.1 |
| 900  | 3*898               | 1.3e-7                  | 14.4 | 2*899        | 3.4e-5                  | 1.4 | 2*899                  | 0.5e-9                  | 5.6 |
| 1000 | 3*998               | 2.6e-7                  | 16.0 | 2*999        | 3.8e-5                  | 1.6 | 2*999                  | 4.6e-9                  | 5.9 |

表 1: テスト問題に対する結果

- chained singular 関数 [1]:

$$f_0(x) = \sum_{i \in J} ((x_i + 10x_{i+1})^2 + 5(x_{i+2} - x_{i+3})^2 + (x_{i+1} - 2x_{i+2})^4 + 10(x_i - 10x_{i+3})^4)$$

ただし,  $J = \{1, 3, 5, \dots, n-3\}$  で  $n$  は 4 の倍数.

- Broyden banded 関数 [9]:

$$f_0(x) = \sum_{i=1}^n \left( x_i(2 + 5x_i^2) + 1 - \sum_{j \in J_i} (1 + x_j)x_j \right)^2$$

ただし,  $J_i = \{j \mid j \neq i, \max(1, i-5) \leq j \leq \min(n, i+1)\}$ .

数値実験は以下の表 2 のようになった. 表 2 において,  $\epsilon_{\text{obj}}$  は, 疎性を利用する半正定値計画

| Chained singular function |        |                         |      |        | Broyden banded function |        |                         |       |        |
|---------------------------|--------|-------------------------|------|--------|-------------------------|--------|-------------------------|-------|--------|
| $n$                       | cl.str | $\epsilon_{\text{obj}}$ | (9)  | (8)    | $n$                     | cl.str | $\epsilon_{\text{obj}}$ | (9)   | (8)    |
| 16                        | 3*14   | 3.5e-7                  | 0.6  | 3059.5 | 6                       | 6*1    | 8.0e-9                  | 11.3  | 11.6   |
| 40                        | 3*38   | 9.0e-7                  | 1.4  | —      | 7                       | 7*1    | 1.9e-8                  | 69.5  | 69.5   |
| 100                       | 3*98   | 7.8e-7                  | 3.8  | —      | 8                       | 7*2    | 2.8e-8                  | 164.1 | 373.7  |
| 200                       | 3*198  | 5.4e-7                  | 8.4  | —      | 9                       | 7*3    | 9.1e-8                  | 240.3 | 1835.6 |
| 400                       | 3*398  | 3.4e-7                  | 19.3 | —      | 10                      | 7*4    | 6.2e-8                  | 348.7 | 8399.4 |

表 2: chained singular 関数と Broyden banded 関数に対する結果

緩和による最適値の精度を表しており, (9) の列は, 疎性を利用する半正定値計画緩和による CPU 時間を表している. また, (8) の列は, 疎性を利用しない半正定値計画緩和による CPU 時間表している. この表から, 疎性を利用することで, 非常に高速に精度の良い最適値が得られることが分かる.

#### 6.4 制約式がある場合

テスト問題として, [5] にある問題の中から数問を選んで, 疎性を利用する半正定値計画緩和と利用しない半正定値計画緩和を適用した.

| 問題名           | $n$ $r$ |   | (12)             |                   |                    |        | (13)             |                    |      |
|---------------|---------|---|------------------|-------------------|--------------------|--------|------------------|--------------------|------|
|               |         |   | $\epsilon_{obj}$ | $\epsilon_{feas}$ | $\epsilon'_{feas}$ | cpu    | $\epsilon_{obj}$ | $\epsilon'_{feas}$ | cpu  |
| ex5_2_2_case1 | 9       | 2 | h.ac             | -2e+1             | -9e-5              | 1.0    | h.ac             | h.ac               | 1.6  |
| ex5_2_2_case1 | 9       | 3 | 4e-8             | -7e-6             | -4e-9              | 138.3  | —                | —                  | —    |
| ex5_2_2_case2 | 9       | 2 | h.ac             | -7e+1             | -3e-4              | 0.9    | 2e-9             | h.ac               | 1.4  |
| ex5_2_2_case2 | 9       | 3 | h.ac             | -4e-4             | -2e-9              | 131.9  | —                | —                  | —    |
| ex5_2_2_case3 | 9       | 2 | h.ac             | -6e+1             | -2e-4              | 0.8    | 3e-8             | -3e-9              | 1.0  |
| ex5_2_2_case3 | 9       | 3 | h.ac             | -3e-3             | -1e-8              | 186.2  | —                | —                  | —    |
| ex9_1_1       | 13      | 2 | h.ac             | -7e-9             | h.ac               | 0.7    | h.ac             | h.ac               | 2.7  |
| ex9_1_2       | 10      | 2 | h.ac             | -5e-8             | h.ac               | 0.5    | h.ac             | h.ac               | 1.0  |
| ex9_2_1       | 10      | 2 | 2e-9             | -9e-9             | h.ac               | 0.5    | h.ac             | h.ac               | 0.8  |
| ex9_2_2       | 10      | 2 | 5e-6             | -2e-5             | -4e-8              | 0.5    | 5e-6             | -4e-8              | 1.1  |
| ex9_2_3       | 16      | 2 | 4e-7             | -4e-2             | -7e-6              | 0.8    | 1e-7             | -4e-6              | 14.1 |
| alkyl         | 14      | 2 | 1e-2             | -7e-1             | -1e-1              | 1.8    | 7e-3             | -1e+1              | 14.4 |
| alkyl         | 14      | 3 | h.ac             | h.ac              | h.ac               | 1923.1 | —                | —                  | —    |
| st_jcbpaf2    | 10      | 2 | h.ac             | h.ac              | h.ac               | 1.4    | h.ac             | 0e+0               | 1.3  |

表 3: [5] にある問題に対する結果

なお, 表中で h.ac は,  $1.0 * 10^{-9}$  より小さな値であることを意味している. また,  $\epsilon'_{feas}$  は, (1) に対して, スケーリングした最適化問題に対する  $\epsilon_{feas}$  の値である. 表 3 では, (12) の列が疎性を利用しない半正定値計画緩和の, (13) の列が疎性を利用する半正定値計画緩和の結果である. 疎性を利用する場合, 多くの問題を高速に解いているが, ex5\_2\_2\_2\_case1 の様に, 疎性を利用しない方が高速に求められる場合もあるのがわかる.

## 7 おわりに

本稿では, まず, Lasserre によって提案された, 多項式最適化問題に対する半正定値計画緩和と, 疎性を利用して高速に解く手法を提案した. 数値実験をまとめた表を見ても分かるように, 疎性のある多項式最適化問題に対しては, 疎性を利用することで数倍から数十倍高速に解くことができた.

今後の課題は, 収束性の証明や, 等式が入っている場合に, 得られる最適解の精度が悪い場合があるので, その改善する必要がある. また, 提案した疎性を持たない場合の工夫も考える必要がある.

## 参考文献

- [1] A. R. Conn, N. I. M. Gould and P. L. Toint, "Testing a class of methods for solving minimization problems with simple bounds on the variables", *Math. Comp.*, **50** (1988) 399-430

- [2] C. Floudas, P. Pardalos, C. Adjiman, W. Esposito, Z. Gümmüs, S. Harding, J. Klepeis, C. Meyer and C. Schweiger *Handbook of test problems in local and global optimization* Kluwer Academic Publishers, 1999
- [3] M. Fukuda, M. Kojima, K. Murota and K. Nakata, "Exploiting sparsity in semidefinite programming via matrix completion I: General framework," *SIAM J. Optim.*, **11** (2000) 647-674.
- [4] D. R. Fulkerson and O. A. Gross, "Incidence matrices and interval graphs", *Pacific J. Math.*, **15** (1965) 835-855.
- [5] GLOBAL Library, <http://www.gamsworld.org/global/globallib.htm>
- [6] S. Kim, M. Kojima and H. Waki, "Generalized Lagrangian duals and sums of squares relaxations of sparse polynomial optimization problems", *SIAM J. Optim.*, **15** (2005) 697-719 .
- [7] M. Kojima, S. Kim and H. Waki, "Sparsity in sums of squares of polynomials", *Math. Program.*, **103** (2005) 45-62.
- [8] J. B. Lasserre, "Global optimization with polynomials and the problems of moments", *SIAM J. Optim.*, **11** (2001) 796-817.
- [9] J. J. More, B. S. Garbow and K. E. Hillstom, "Testing Unconstrained Optimization Software", *ACM Trans. Math. Soft.*, **7**, (1981) 17-41
- [10] S. G. Nash, "Newton-Type Minimization via the Lanczos method", *SIAM J. Numer. Anal.*, **21** (1984) 770-788.
- [11] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems", *Mathematical Programming*, **96** (2003) 293-320.
- [12] B. Reznick, "Some concrete aspects of Hilbert's 17th problem", *Proc. of Contemporary Mathematics*, (2000) 251-272.
- [13] J. F. Strum, "SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones", *Optim. Meth. Softw.*, **11 & 12** (1999) 625-653.
- [14] H. Waki and S. Kim and M. Kojima and M. Muramatsu, "Sums of Squares and Semidefinite Programming Relaxations for Polynomial Optimization Problems with Structured Sparsity" newblock Research Report B-395, Dept. of Mathematical and computing Sciences, Tokyo Institute of Technology, Meguro, Tokyo 152-8552.
- [15] M. Yamashita, K. Fujisawa and M. Kojima, "Implementation and evaluation of SDPA 6.0 (SemiDefinite Programming Algorithm 6.0)", *Optim. Meth. Softw.*, **18** (2003) 491-505.